

Aufgabe 1

Ändern Sie die Syntax von WHILE, indem Sie INTEGER- und REALZahlen unterscheiden.
In der Vorlesung wurden die ganzen Zahlen wie folgt definiert:

```
1 // ganze Zahlen (endlicher Ausschnitt der ganzen Zahlen MIN+1 .. MAX)
2 Z ::= 0 | 1 | ... | MAX | -1 | -2 | ... | MIN
```

Gleitkommazahlen können in WHILE so abgebildet werden:

```
1 // reelle Zahlen
2 R ::= MAX | ... | 2/1 | 1/1 | 1/2 | ... | 0 | ... | -1/2 | -1/1 | -2/1 | ... | MIN
```

Jedoch ist hier die Null, sowie MAX und MIN doppelt definiert. Eine bessere Lösung ist demnach folgende:

```
1 // ganze Zahlen (endlicher Ausschnitt der ganzen Zahlen MIN+1 .. MAX)
2 Z ::= 1 | 2 | ... | MAX | -1 | -2 | ... | MIN
3 // reelle Zahlen
4 R ::= Z/Z | 0
```

Die Definition der reellen Zahlen muss zur Vollständigkeit in die Konstanten und Terme eingetragen werden.

```
1 K ::= Z | R | W
2 T ::= Z | R | I | T1 OP T2 | read, für T1, T2 in TERM
```

Aufgabe 2

Definieren Sie für eine geeignete Erweiterung der Sprache WHILE eine konkrete Syntax, die eindeutig ist.
Zusätzlich zum If-Else wäre Switch:

```
1 C ::= switch T: case B: do C
```

Eine weitere wirklich sinnvolle Erweiterung wäre das Programm. Es besteht aus einem oder mehreren Befehlen.

```
1 //Programm
2 P ::= C
```

Aufgabe 3

Formulieren Sie informell eine Präzisierung der angegebenen WHILE-Semantik, die die genannten Fehlerquellen:

Bereichsüberschreitungen Man verlässt den definierten Bereich der ganzen Zahlen, also MIN-1 oder MAX+1.

Division durch Null

Berechnung von read bei leerer Eingabedatei Es müsste einen leeren Rückgabewert in den Konstanten geben.

Typkonflikte Es werden zwei nicht kompatible Typen miteinander verglichen.
behandelt.